

```
CREATE EXTENSION postgis;
```

```
CREATE TABLE public.alembic_version (  
  version_num text NOT NULL,  
  CONSTRAINT alembic_version_pkc PRIMARY KEY (version_num)  
);
```

```
CREATE TABLE public.region (  
  id_region bigint primary key generated always as identity,  
  nombre_region text NOT NULL  
);
```

```
CREATE TABLE public.usuario (  
  id_usuario text NOT NULL,  
  correo text NOT NULL,  
  clave text NOT NULL,  
  tipo_usuario text NULL,  
  activo boolean NOT NULL,  
  nombres text NOT NULL,  
  apellidos text NOT NULL,  
  telefono text NULL,  
  fecha_creacion timestamp with time zone NOT NULL,  
  imagen text NULL,  
  links_contacto json NULL,  
  refresh_token text NULL,  
  refresh_token_expiry timestamp with time zone NULL,  
  CONSTRAINT usuario_pkey PRIMARY KEY (id_usuario),  
  CONSTRAINT usuario_correo_key UNIQUE (correo)  
);  
CREATE INDEX IF NOT EXISTS idx_usuario_correo ON public.usuario USING btree (correo);
```

```
CREATE TABLE public.ciudad (  
  id_ciudad bigint primary key generated always as identity,  
  nom_ciudad text NOT NULL,  
  id_region bigint NOT NULL,  
  CONSTRAINT ciudad_id_region_fkey FOREIGN KEY (id_region) REFERENCES  
public.region(id_region)  
);
```

```
CREATE TABLE public.comuna (  
  id_comuna bigint primary key generated always as identity,  
  nom_comuna text NOT NULL,  
  id_ciudad bigint NOT NULL,
```

```
id_region bigint NOT NULL,  
CONSTRAINT comuna_id_ciudad_fkey FOREIGN KEY (id_ciudad) REFERENCES  
public.ciudad(id_ciudad),  
CONSTRAINT comuna_id_region_fkey FOREIGN KEY (id_region) REFERENCES  
public.region(id_region)  
);
```

```
CREATE TABLE public.vecindario (  
id_vecindario bigint primary key generated always as identity,  
nom_vecindario text NOT NULL,  
id_comuna bigint NOT NULL,  
id_ciudad bigint NOT NULL,  
id_region bigint NOT NULL,  
CONSTRAINT vecindario_id_ciudad_fkey FOREIGN KEY (id_ciudad) REFERENCES  
public.ciudad(id_ciudad),  
CONSTRAINT vecindario_id_comuna_fkey FOREIGN KEY (id_comuna) REFERENCES  
public.comuna(id_comuna),  
CONSTRAINT vecindario_id_region_fkey FOREIGN KEY (id_region) REFERENCES  
public.region(id_region)  
);
```

```
CREATE TABLE public.vivienda (  
id_vivienda text NOT NULL,  
area_total double precision NULL,  
pisos integer NULL,  
habitaciones integer NULL,  
precio_uf double precision NOT NULL,  
estaciona integer NULL,  
bodega integer NULL,  
antiguedad integer NULL,  
tipo_vivienda integer NOT NULL,  
nombre_propiedad text NOT NULL,  
descripcion text NOT NULL,  
condicion text NULL,  
tipo_operacion boolean NULL,  
banos integer NULL,  
area_construida double precision NULL,  
latitud double precision NOT NULL,  
longitud double precision NOT NULL,  
ubicacion geometry NOT NULL,  
tipo_subsidio text NULL,  
fecha_creacion timestamp with time zone NOT NULL,  
links_contacto json NULL,
```

```

id_vecindario bigint NULL,
id_comuna bigint NOT NULL,
id_ciudad bigint NOT NULL,
id_region bigint NOT NULL,
id_usuario text NULL,
CONSTRAINT vivienda_pkey PRIMARY KEY (id_vivienda),
CONSTRAINT vivienda_id_ciudad_fkey FOREIGN KEY (id_ciudad) REFERENCES
public.ciudad(id_ciudad),
CONSTRAINT vivienda_id_comuna_fkey FOREIGN KEY (id_comuna) REFERENCES
public.comuna(id_comuna),
CONSTRAINT vivienda_id_region_fkey FOREIGN KEY (id_region) REFERENCES
public.region(id_region),
CONSTRAINT vivienda_id_usuario_fkey FOREIGN KEY (id_usuario) REFERENCES
public.usuario(id_usuario),
CONSTRAINT vivienda_id_vecindario_fkey FOREIGN KEY (id_vecindario) REFERENCES
public.vecindario(id_vecindario)
);
CREATE INDEX IF NOT EXISTS idx_vivienda_id_vecindario ON public.vivienda USING btree
(id_vecindario);
CREATE INDEX IF NOT EXISTS idx_vivienda_ubicacion ON public.vivienda USING gist
(ubicacion);

```

```

CREATE TABLE public.imagen (
id_imagen text NOT NULL,
url text NOT NULL,
id_vivienda text NOT NULL,
CONSTRAINT imagen_pkey PRIMARY KEY (id_imagen),
CONSTRAINT imagen_id_vivienda_fkey FOREIGN KEY (id_vivienda) REFERENCES
public.vivienda(id_vivienda)
);

```

```

CREATE TABLE public.favorito (
id_favorito text NOT NULL,
fecha_guardado timestamp with time zone NOT NULL,
id_usuario text NOT NULL,
id_vivienda text NOT NULL,
CONSTRAINT favorito_pkey PRIMARY KEY (id_favorito),
CONSTRAINT favorito_id_usuario_fkey FOREIGN KEY (id_usuario) REFERENCES
public.usuario(id_usuario),
CONSTRAINT favorito_id_vivienda_fkey FOREIGN KEY (id_vivienda) REFERENCES
public.vivienda(id_vivienda)
);
CREATE INDEX IF NOT EXISTS idx_favorito_id_usuario ON public.favorito USING btree

```

```
(id_usuario);  
CREATE INDEX IF NOT EXISTS idx_favorito_id_vivienda ON public.favorito USING btree  
(id_vivienda);
```

```
CREATE TABLE public.match (  
  id_match text NOT NULL,  
  fecha_coincidencia timestamp with time zone NOT NULL,  
  visto boolean NOT NULL,  
  id_usuario text NOT NULL,  
  id_vivienda text NOT NULL,  
  CONSTRAINT match_pkey PRIMARY KEY (id_match),  
  CONSTRAINT match_id_usuario_fkey FOREIGN KEY (id_usuario) REFERENCES  
public.usuario(id_usuario),  
  CONSTRAINT match_id_vivienda_fkey FOREIGN KEY (id_vivienda) REFERENCES  
public.vivienda(id_vivienda)  
);  
CREATE INDEX IF NOT EXISTS idx_match_id_usuario ON public.match USING btree  
(id_usuario);  
CREATE INDEX IF NOT EXISTS idx_match_id_vivienda ON public.match USING btree  
(id_vivienda);
```

```
CREATE TABLE public.preferencia (  
  id_preferencia text NOT NULL,  
  area_total double precision NULL,  
  pisos integer NULL,  
  habitaciones integer NULL,  
  estaciona integer NULL,  
  bodega integer NULL,  
  antiguedad integer NULL,  
  tipo_vivienda integer NULL,  
  condicion text NULL,  
  tipo_operacion boolean NULL,  
  banos integer NULL,  
  area_construida double precision NULL,  
  tipo_subsidio json NULL,  
  busqueda_automatica boolean NULL,  
  distancia double precision NULL,  
  contactado boolean NOT NULL,  
  id_usuario text NOT NULL,  
  tipo_valor text NULL,  
  precio_minimo double precision NULL,  
  precio_maximo double precision NULL,  
  notificaciones boolean NULL,
```

```
CONSTRAINT preferencia_pkey PRIMARY KEY (id_preferencia),
CONSTRAINT preferencia_id_usuario_fkey FOREIGN KEY (id_usuario) REFERENCES
public.usuario(id_usuario)
);
CREATE INDEX IF NOT EXISTS idx_preferencia_distancia ON public.preferencia USING btree
(distancia);
CREATE INDEX IF NOT EXISTS idx_preferencia_id_usuario ON public.preferencia USING
btree (id_usuario);
```